3.**Power Law Transform**

Explanation: Power law is used to enhance the contrast of the images

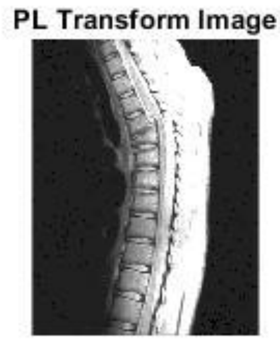
Power law is a functional relationship between two quantities, where a relative change in one quantity results in a proportional relative change in the other quantity, independent of the initial size of those quantities: one quantity varies as a power of another.

Implementing the Power Law Transform: (See the file Powlaw.m)

```
Equation for power law ::   g  =  a*f.^k    (a=constant, k<1 or k>1)
                                            (f is input image, g is output image)
```

Reading input images:

a. **Spiral.tif**
b. **Aerial.tif**

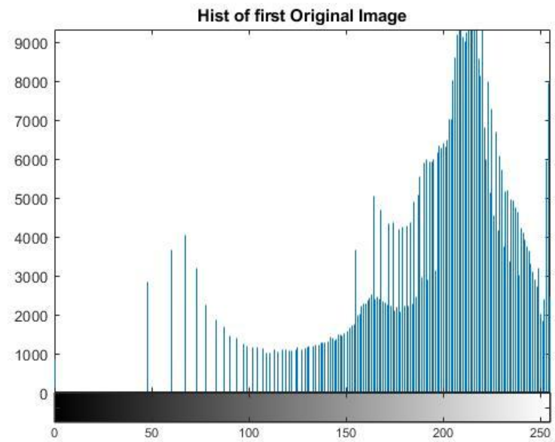


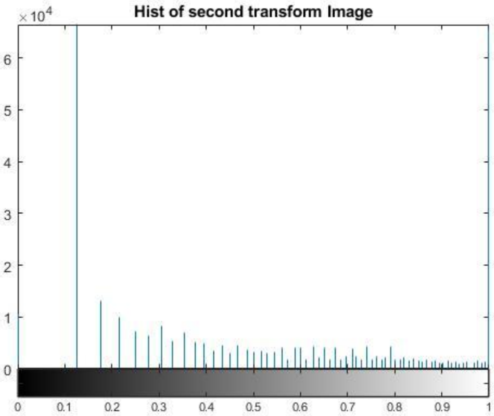
Function working:

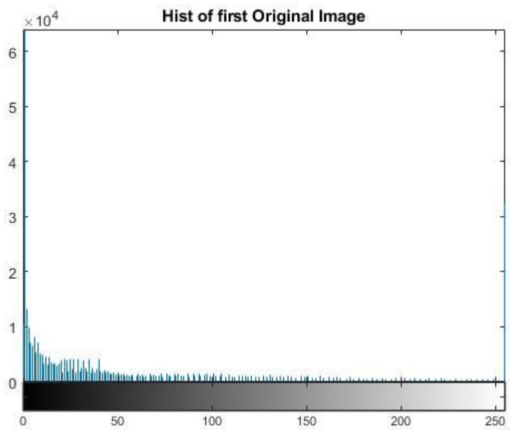
1. Read input image.
2. Take the power of the each pixel value. Here observe the range of k for input images.
3. Multiply it with a constant for scaling.
4. Store it as the output. Display now.

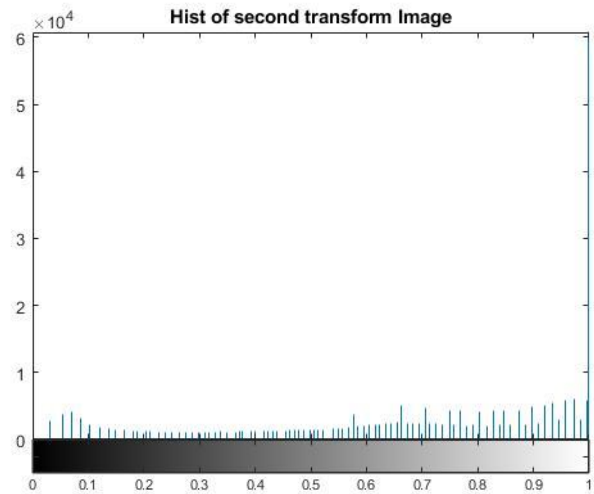
Note: Adjust k value from the function
- if k<1 brightness increases.
- if k>1 brightness decreases.

Outputs:









3.2 Contrast Stretching

Explanation:

It is a simple image enhancement technique that attempts to improve the **contrast** in an image by `**stretching**' the range of intensity values it contains to span a desired range of values, e.g. the the full range of pixel values that the image type concerned allows.

Implementing the Contrast stretching: (See the file histstretch.m)

Equation for Contrast Stretching: o = 255*((I-a)/b-a)

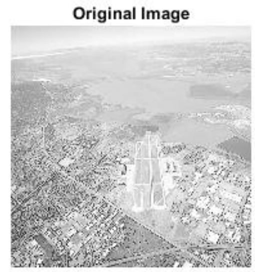
I – Image input
a- Minimum
b- Maximum
o- output image

Reading input images:

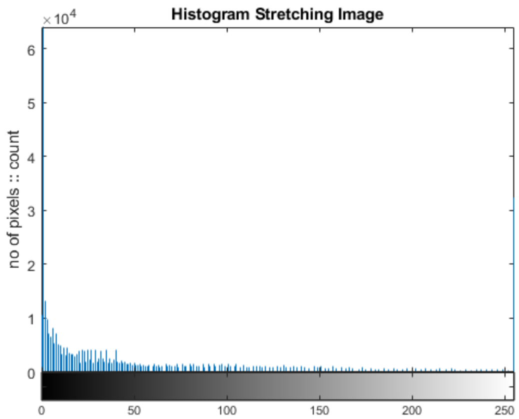
a. **Spiral.tif**
b. **Aerial.tif**

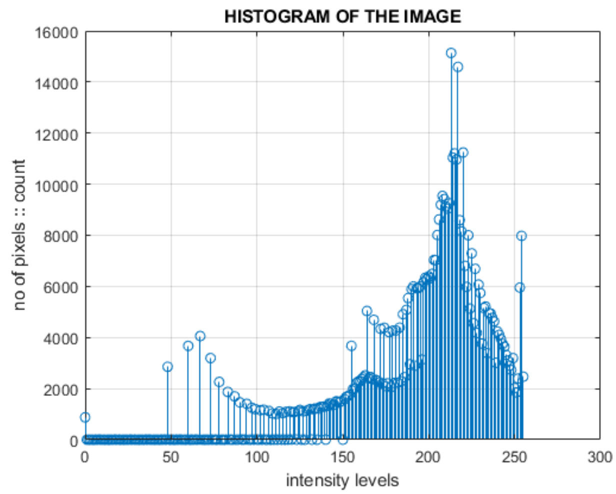
Function working:

1. **Read input image.**
2. **Take the input image, find minima and maxima.**
3. **If o=minima assign ‘0’ and o=maxima assign ‘255’ as output**
4. **Elseif Use the formula above mentioned.**
5. **Store it as the output. Display now.**
6. **Compare it inbuilt imadjust.**



Results for spine and aerial images.





3.3 Histogram Equalization

Explanation:

This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

Implementing the Histogram Equalization: (See the file histeq.m)

Equation for Contrast Stretching: o = 256*(cumulative sum of intensities of ‘I’ image)/(NM)

I – Image input
NM – Normalization factor
o- output image

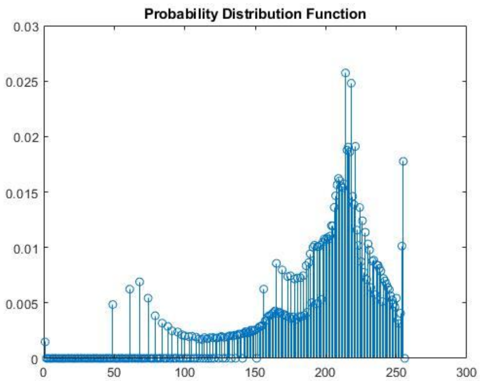
Reading input images:

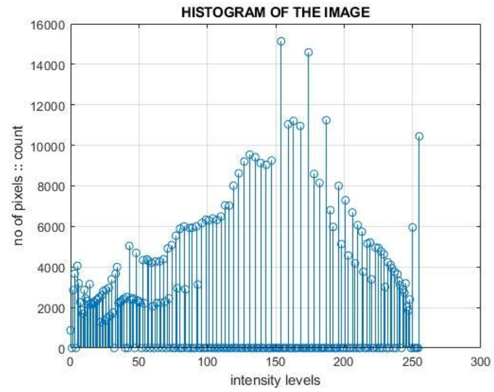
a. **Spiral.tif**
b. **Aerial.tif**

Function working:

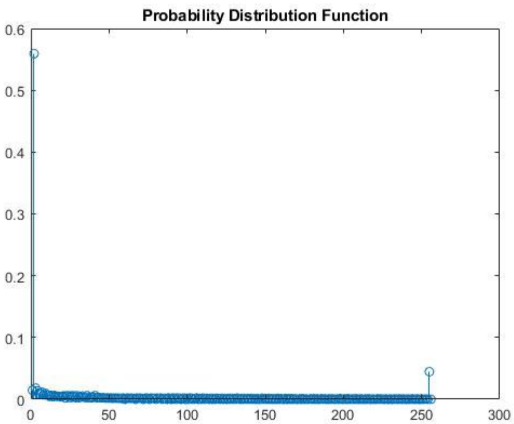
1. **Read input image.**
2. **Take the input image, find no of pixels with same intensities**
3. **Find the probability density function and then find cumulative density function.**
4. **Normalize the pixel values. Round to nearest possible value.**
5. **Store it as the output. Display now.Compare it with imhist.**

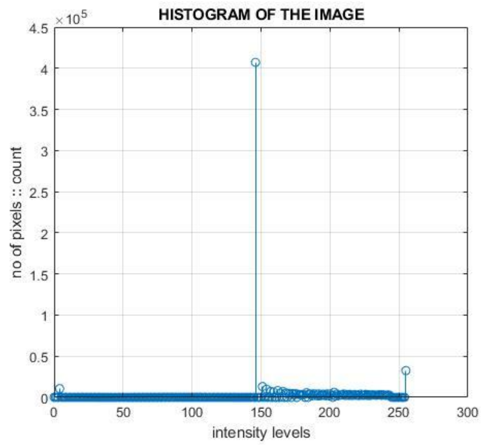
Results for spine image:



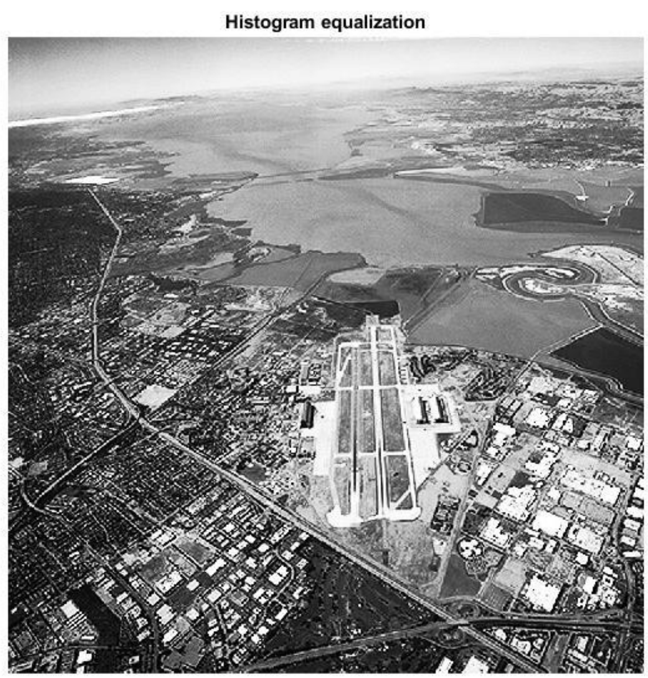


Results for aerial image:





Outputs : Histogram Equalization for spine image and Aerial image



Note2 for power Law transform

--

If k< 1

The range of dark pixels in the image would be expanded and range of bright pixels would be compressed. It should be used on an image in which the region of your interest lies in the dark region or details are covered by dark regions.

If k > 1

The range of dark pixels in the image would be compressed and range of bright pixels would be expanded. It should be used on an image which has a washed-out look or overexposed image. Power law transformation is applied to image () and for increasing gamma values, contrast is enhanced.

```
Note:
Also see the main common dsplab1.m file to call both functions :: to give inputs
generalized way. And display outputs
```

```
%finding power law transforms

b1=imread('spine.tif');

 ib1=im2double(b1);
[r,c]=size(ib1);
o1=ib1;
 % to make image dark take value of gamma > 1, to make image bright take value of gamma < 1
for i=1:r
    for j=1:c
         o1(i,j)=2*(ib1(i,j).^0.3);  % formula to implement power law transformation
    end
end

subplot(2,2,1), imshow(b1);
title('Original Image');
subplot(2,2,2), imshow(o1);
title('PL Transform Image');

b2=imread('aerial.tif');
ib2=im2double(b2);
[r,c]=size(ib2);
o2=ib2;
 % to make image dark take value of gamma > 1, to make image bright take value of gamma < 1
for i=1:r
    for j=1:c
         o2(i,j)=2*(ib2(i,j).^1.8);  % formula to implement power law transformation
    end
end

subplot(2,2,3), imshow(b2);
title('Original Image');
subplot(2,2,4), imshow(o2);
title('PL Transform Image');

figure; imhist(b1);
title('Hist of first Original Image');
figure; imhist(o1);
title('Hist of second transform Image');

figure; imhist(b2);
title('Hist of first Original Image');
figure; imhist(o2);
title('Hist of second transform Image');
```

Published with MATLAB® R2018a

```
function [O] = histstretch(img1)
```

```
%This is a histretch function
%give the input image at the console to display the output
```

```
img1=im2double(img1);
[r,c]=size(img1);
a=min(img1(:));           %minimum intensity of the image
b=max(img1(:));           %maximum intensity of the image
O=zeros(r,c);

for i=1:r
    for j=1:c
        if img1(i,j)<a
            O(i,j)=0;
        elseif img1(i,j)>b
            O(i,j)=255;
            else
              O(i,j)= 255*(img1(i,j)-a)/(b-a);    % implementation of the function
            end
     end
end

subplot(1,2,1), imshow(img1);
title('Original Image');
subplot(1,2,2), imshow(uint8(O));
title('final Image');

figure,imhist(uint8(O));
title('Histogram Stretching Image');

xlabel('intensity levels')
ylabel('no of pixels :: count')
```

```
Not enough input arguments.

Error in histstretch (line 9)
img1=im2double(img1);
```

```
end
```

```
function [H] = histequ(G)
```

```
%This is a histequ function
%give the input image at the console to display the output
```

```
numofpix=size(G,1)*size(G,2);
figure,imshow(G);
title('Original Image');
H=uint8(zeros(size(G,1),size(G,2)));
f=zeros(256,1);
probf=zeros(256,1);
probc=zeros(256,1);
cum=zeros(256,1);
output=zeros(256,1);
n=1:256;
%freq counts the occurrence of each pixel value.
%The probability of each occurrence is calculated by probf.
for i=1:size(G,1)
    for j=1:size(G,2)
        val=G(i,j);
        f(val+1)=f(val+1)+1;
        probf(val+1)=f(val+1)/numofpix;
    end
end
figure,stem(n,probf,'Marker','None')
title('Probability Distribution Function')
sum=0;
no=255;
%The cumulative distribution probability is calculated.
for i=1:size(probf)
   sum=sum+f(i);
   cum(i)=sum;
   probc(i)=cum(i)/numofpix;
   output(i)=round(probc(i)*no);
end


for i=1:size(G,1)
    for j=1:size(G,2)
            H(i,j)=output(G(i,j)+1);
    end
end
figure, imshow(H);
title('Histogram equalization');
figure,histogr(H);
xlabel('intensity levels')
ylabel('no of pixels :: count')
```

```
Not enough input arguments.

Error in histequ (line 9)
numofpix=size(G,1)*size(G,2);
```

```
end
```

```
%Save this file :: it is the main function
%here to run multiple functions histoequi and histostretch
%You can change input images here
%So as to get histoequi and histostretch plots
```

```
im1 = imread('spine.tif');
im2 = imread('aerial.tif');

img1 = imresize(im1,0.5);
img2 = imresize(im2,0.5);

figure(1);
imshow(img1); title ('original img')

figure(2);
imshow(img2); title ('original img')

figure(3);
histstrech1 = histstretch(img1);

figure(4);
histstrech2 = histstretch(img2);

figure(5);
img1_histo = histo(img1); title ('histogram img1')
img1_hist_eq = histequa(img1);
figure(6);

img2_histo = histo(img2); title ('histogram img2')
img2_hist_eq = histequa(img2);
figure(7);
```

```
Undefined function 'histequa' for input arguments of type 'uint8'.

Error in dsplab1 (line 27)
img1_hist_eq = histequa(img1);
```

Published with MATLAB® R2018a